基于目标的域随机化方法在机器人操作方面的研究 *

张夏禹†，陈小平
(中国科学技术大学，合肥 230026)

摘　要：使用强化学习解决机器人操作问题有着诸多优势，然而传统的强化学习算法面临着奖励稀疏的困难，且得到的策略难以直接应用到现实环境中。为了提高策略从仿真到现实迁移的成功率，提出了基于目标的域随机化方法：使用了基于目标的强化学习算法对模型进行训练，可以有效的应对机器人操作任务奖励稀疏的情况，得到的策略可以在仿真环境下良好运行，于此同时在算法中还使用了目标驱动的域随机化的方法，在提高策略泛用性以及克服仿真和现实环境之间的差距上有着良好的效果，仿真环境下的策略容易迁移到现实环境中并成功执行。结果表明，使用了基于目标的域随机化方法的强化学习算法有助于提高策略从仿真到现实迁移的成功率。

关键词：强化学习；域随机化；机器人操作；仿真到现实迁移
中图分类号：TP399　　doi: 10.19734/j.issn.1001-3695.2022.03.0108

Research on goal-based domain randomization method in robot manipulation

Zhang Xiayu†, Chen Xiaoping
(*University of Science & Technology of China*, *Hefei* 230026, *China*)

Abstract: Reinforcement learning method has many advantages in solving the robot manipulation problems. However, the traditional reinforcement learning algorithm faces the difficulty of sparse reward, and the policy is difficult to be directly applied to the reality. In order to improve the success rate of policy migration from simulation to reality, this paper proposes a goal-based domain randomization method: The method uses the goal-based reinforcement learning algorithm to train the model, which can effectively deal with the sparse reward of robot manipulation tasks, and the policy can run well in the simulation environment. At the same time, the method uses the goal-conditioned domain randomization algorithm, which has a good performance on improving the universality of policy and overcoming the reality gap between simulation and reality. The policy in simulation is easy to migrate to reality and execute successfully. The results show that the reinforcement learning algorithm using the goal-based domain randomization method helps to improve the success rate of policy migration from simulation to reality.

Key words: reinforcement learning; domain randomization; robot manipulation; sim-to-real

0　引言

随着人工智能技术的发展，自动化设备的普及，机器人操作在现实生活中扮演着愈发重要的角色[1]。不同于传统的任务规划方法，强化学习使得智能体通过与环境的交互，根据奖励函数的反馈，自主发现最优策略，而不需要设计者去关心解决问题的具体细节[2]，因此在解决机器人操作问题方面有着天然的优势：例如 OpenAI 的团队已经在机械灵巧手上实现了复杂的操作[3]，而国内团队也在 Kinova 机械臂上利用强化学习实现了物体的抓取[4]。

机器人操作的问题的任务空间巨大，奖励稀疏，而复杂操作任务又难以直接人工定义奖励函数，因此在实际应用的场景中，基于强化学习的算法仍旧面临着巨大的挑战。

另一方面，在实际操作任务中使用强化学习也面临着重重困难，直接在现实环境中采样训练难以执行，比如：采样效率太低，训练和测试的过程对人员和设备都十分危险。将仿真环境中学习训练得到的策略直接应用在现实场景中看似可行，但因为仿真器对于现实物理环境建模的误差和仿真环境获取数据与实际采样时会产生的误差和干扰会导致策略无法使用。因此，解决仿真环境和现实世界环境之间的不匹配，克服“现实差距”(Reality Gap)带来的影响，是通过强化学习解决机器人操作任务的发展方向。

关于如何解决这个现实差距的问题，目前存在着许多种方法。将控制策略从仿真转移到现实世界的问题可以看做是域自适应的一个实例，在源域中训练的模型被转移到新的目标域。这些方法基于一个关键的假设：不同的域具有共同的特征，因此在其中一个域中学习到的表征和行为在另一个域中也可以利用。其中，域适应方法(Domain Adaption)通过学习一个模拟环境和现实环境共同的状态到隐变量空间的映射，在模拟环境中，使用映射后的状态空间进行算法的训练；在迁移到现实环境中时，同样将状态映射到隐含空间后即可直接应用在模拟环境训练好的模型[5]。而域随机化(domain randomization)的方法则是对模拟环境中的信息或者参数进行随机化。从理论的角度，陈骁宇等人[6]对仿真到现实迁移这一经典问题给出一个理论的解释模型，尤其是对域随机化算法为什么有效和在什么场景下有效等问题给出理论解释。通过 POMDP 模型的论证，证明了域随机化对于解决机器人操作领域的仿真到现实的迁移问题有着良好的效果。并证明了域随机化方法有着出色的性能保证，优秀的设计理论上可以实现不使用任何真实场景数据的训练。Peng 等人[7]通过随机化物理参数的方式，将智能体在大量不同物理参数确定的虚拟环境中优化累积回报的期望值，试图使训练出的策略更

收稿日期：2022-03-07；**修回日期：**2022-05-06　**基金项目：**国家重点研发计划项目(2019YFE0125200)
作者简介：张夏禹(1994-)，男(回族，通信作者)，天津人，硕士研究生，主要研究方向为强化学习，机器人操作(pb120110@mail.ustc.edu.cn)；陈小平(1955-)，男，北京人，中国科学技术大学计算机科学与技术学院教授、博导、机器人实验室主任、人工智能中心主任，主要研究方向为人工智能与机器人的交叉研究和教学，自然语言理解和自动推理.

加鲁棒。Chebotar 等人[8]以此为基础在域随机化物理参数之后，利用现有策略在虚拟环境和现实中对于相同的初始状态分别产生一条轨迹，通过比较两条轨迹的差距修正随机化的物理参数。Tobin 等人[9]则使用域随机化的方法随机化了环境的视觉表示。Niu 等人[10]使用域随机化方法提升了自动驾驶在仿真环境下训练的鲁棒性。

上述方法在关注源域和目标域即仿真和现实差异的时候重视了客观环境上的差异(一般是物理参数的差异)，而对于环境中驱动的实体差异关注较少。事实上机器人操作本身是一个欠驱动的系统，对于系统实际控制的结果往往和模型预期的也有差距，这个差异主要是由驱动器的驱动方式和传感器反馈的偏差造成的，和环境参数的误差关系不大，使用精度更高的驱动器可以缩小这个差距但是无法完全抹除，对于环境参数的随机化在这个方面往往无能为力。

除了域相关的方法外，Andrei 等人[11]则通过将一类特殊的渐进式神经网络(progressive neural network)扩展到强化学习中来训练模型。Christiano 等人[12]则是利用逆动力学模型(inverse dynamic model)的方法。然而这些方法比较依赖模型，在不同操作任务之间的泛化性能较为一般。

综上所述，目前的方法在克服仿真和现实之间的差异方面各有长处，然而都难以保证迁移成功率的同时兼顾训练的速度和算法在不同任务中的泛用性。因此本文提出了一种基于目标的域随机化方法，通过经验回放的方法解决了强化学习中奖励稀疏的问题，同时通过域随机化方法提高了策略对于现实环境和仿真环境差异的适应能力，不仅在训练效率上优于其他域随机化算法，还保证了在现实环境执行任务的时候有着较高的成功率。

1 机器人操作问题

1.1 强化学习的机器人操作任务

根据上述内容，本文首先将机器人操作的问题描述为一个强化学习问题[13]。在机器人操作任务中，通常可以将一个标准的强化学习模型描述为：一个智能体(agent)通过与环境交互来使得回报最大化的过程，为了便于接下来的描述，本文假设问题的环境是可完全观测的。一个确定的策略 $\pi(a|s)$ 是从状态 S 到行为 A 的映射，对策略的每个查询都会从特定分布中对操作进行采样。奖励函数 r : S×A→R 则返回一个值，表示的是在给定状态下执行特定操作的价值。状态转移概率 $p(s_{t+1}|s_t,a_t)$ 表示的是状态 s_t 执行动作 a_t 后转移到状态 s_{t+1} 的概率分布。在每个时间步 t，智能体都会根据当前状态从策略中生成一个操作：$a_t=\pi(s_t)$。然后它得到奖励 $r_t=r(s_t,a_t)$，并从状态分布 $p(\cdot|s_t,a_t)$ 中得到新的状态。智能体的目标是最大化其预期回报 ga ，其中 $\gamma\in[0,1]$ 是折扣率。动作价值函数定义为

$$Q_\pi(s,a)=\mathrm{E}[R_t|s_t,a_t] \tag{1}$$

若对于任何 π,s,a，都有一个策略 π^* 使得 $Q_{\pi^*}(s,a)\geq Q_\pi(s,a)$，则将 π^* 称为最优策略。所有最优策略都具有相同的 Q 函数，称为最优 Q 函数并表示为 Q^*。最优函数 Q^* 满足以下贝尔曼方程：

$$Q^*(s,a)=E_{s'\sim p(\cdot|s,a)}[r(s,a)+\gamma maxQ^*(s',a')] \tag{2}$$

1.2 事后经验回放

因为机器人操作的任务空间，奖励比较稀疏，一般的强化学习算法难以收敛。对于大多数强化学习算法来说，从稀疏的二元奖励中学习到成功的策略是一个巨大的挑战。事实上，以简单的二进制数位翻转的任务为例：这个任务中的状态是二进制数序列 $S=\{0,1\}^n$，动作是从 n 个位置中任意挑选一个位置进行翻转。如果奖励函数设置为：当序列正确为 0，否则为-1。则当序列长度达到 20 以上之后，传统的强化学习算法使用二元的奖励函数就已经训练不出结果了。

再以机器人操作任务中最常见的推箱子(Pushing)为例，在常规的仿真环境下，状态空间 S 设置为智能体可以运动到的所有位置，动作空间是一个二维的元组(x,y)代表智能体在 x 和 y 方向上行进的距离，通常步长是 0.01 秒，设置的步长上线是 200 步。一个简单且容易构造的奖励函数二元奖励 r(s，g)，它只返回了给定状态是否满足目标。对于每一步，都会对初始状态和目标状态进行统一采样，只要没有达到目标状态，策略就将获得一个-1 的奖励，即：

$$r(s,a)=\begin{cases}0, & goal\\-1, & otherwise\end{cases} \tag{3}$$

在推箱子任务中，如果使用了上述的奖励函数，则只有当箱子被推到目标位置的时候才能获得正常的奖励，其他的时候都将只获得-1 的奖励，当操作的任务空间比较大的时候，这个奖励将会过于稀疏以至于大部分算法都难以收敛。

对于可以充分建模的任务，强化学习算法可以通过精心设计的奖励函数来引导智能体实现任务的总体目标。例如前面提到的位翻转任务，如果将奖励函数设计为

$$r(s_t,a)=-\|s_{t+1}-g\|^2 \tag{4}$$

则一般的强化学习算也能表现出优异的性能。但是，对于复杂问题，设计奖励函数的难度往往十分巨大，并可能使策略偏向于采用不太理想的行为。现实中机器人操作的任务环境都是复杂的，对于其中的大多数任务来说构造一个专用的奖励函数并不现实。因此，本文利用事后经验回放[14](HER)解决这个问题，目的就是可以使用稀疏奖励和非特定构造的二元奖励函数来训练策略。

事后经验回放(HER)算法[14]基于一个简单的想法：稀疏的奖励空间中进行强化学习训练往往会产生大量的失败轨迹，如果能将这些失败的轨迹利用起来，提高学习的效率，就有可能使用简单的非精心构造的奖励函数来训练一个可行的策略。在一次失败的轨迹中，真正的目标 G 在整个轨迹中都并未实现。而由于失败的轨迹没有实现目标，智能体则完全无法从这样的奖励信息中去更新策略，即整个轨迹在每一个时间步都只有-1 的奖励对于一般的强化学习算法是难以利用的。在回放中，对于没有到达目标的轨迹，提取出这些轨迹已经完成的状态作为虚拟目标，利用这个虚拟目标给出轨迹的奖励同时使用强化学习的方法进行训练。虽然该轨迹在原目标下不成功，但在新的虚拟目标下它将成为成功的轨迹。因此，根据虚拟目标计算的奖励将不会只有-1。通过重现过去的经历，智能体可以用比原始记录轨迹中更成功的例子进行训练。

事后经验回放[14]是一种基于目标[15][16]的强化学习算法，在训练策略的同时，输入的状态不仅包含原本的状态 s，同时加上了一个新的目标 g，相当于此时的状态可以表示为 $s\|g$。定义 $\mathbb{S}$ 为选取新的 g 的策略，这里使用同一轨迹中的某个状态 s 之后的随机 k 个 s 作为目标 g，在设置好目标 g 之后即可利用相应的轨迹生成新的奖励。

在事后经验回放中使用了 DDPG(Deep deterministic policy gradient)算法作为离线(off-policy)强化学习算法[17]。DDPG 是解决连续控制型问题的一个算法，在 DDPG 网络结构中，需要维护两个神经网络：一个 Critic 网络用于对 Q 值进行评估，一个 Actor 网络用于生成目标策略 $\pi(a|s)$。下面是对于整体 HER 算法的描述：

算法 1 事后经验回放(HER)

输入：离线强化学习算法 A；目标选取策略 $\mathbb{S}$；奖励函数 r；

输出：训练好的策略网络

1：初始化 A，回放缓冲区 R

2：for episode in range

3： 获得初始状态 s_0 和目标 g

for t in range (0, T-1)
根据当前状态 s_t 和 g 计算出下一步的动作 a_t
执行动作 a_t 得到新的状态 s_{t+1}
end for
for t in range (0, T-1)
$r_t = r(s_t, a_t, g)$
将轨迹 $(s_t \| g, a_t, r_t, s_{t+1} \| g)$ 存入缓冲区 R 内
新的目标 $G \leftarrow \mathbb{S}$ 用于经验回放
$r' = r(s_t, a_t, g')$
将轨迹 $(s_t \| g', a_t, r', s_{t+1} \| g')$ 存入 R
end for
利用 A 和 R 优化策略
end for

2 基于目标的域随机化算法

然而单纯的经验回放算法得出的策略泛用性较差，在现实环境中难以利用。对于复杂的任务和多变的任务目标，训练难以收敛。并且当前技术无法作出一个能完全模拟现实环境的仿真器，仿真环境和现实环境的差别会导致策略迁移到显示环境中执行时成功率很低。再以推箱子任务为例，仿真环境可以取得准确的物块坐标和机械臂末端坐标，但是在现实环境中重复相同实验的时候，上述坐标的获取只能通过MCS 这类定位系统来实现，而定位系统采集到的数据会受到环境中噪声的干扰，因此存在采样误差。并且显示环境的物理特性难以完全在仿真环境中还原，同样一个状态下执行相同策略相同动作后得到的状态不仅可能与仿真环境下的预测不同，更有可能在每一次执行的时候都有所区别。具体的说，现实环境中采样的频率、延迟，包括执行策略过程中的抖动和环境中无法获取的细微差别，都会对结果产生干扰。

本文使用域随机化(Domain Randomization)方法来解决这个问题：域随机化就是一种互补的适应技术，通过域随机化的方式，源域和目标域之间的差异被建模为源域中的可变性。考虑到现实环境和仿真环境的差异是由多种因素共同构成的，然而最终的结果都会对于执行过程中的任务状态产生影响。因此本文试图通过随机化训练过程中的目标 这种方式，来模拟实际状态和目标之间因为现实差距产生的误差，从而使得得到的策略具有更强的泛用性和鲁棒性。

为了让训练出来的策略有更强的泛化能力，同时在现实中执行时有更高的成功率，本文提出了基于目标的域随机化算法。本文的目标是训练一个策略，可以同时在现实和仿真环境中执行。考虑到现实环境中采样的困难，本文在仿真环境下进行训练，测试其成功率和训练速度，然后将得到的策略迁移到现实环境中进行进一步的测试。考虑到使用的方法，此处的策略使用训练好的神经网络来表达。实际测试的时候通过将采样得到的环境状态输入网络来得到每一步的行动策略。

2.1 域随机化方法

在仿真到现实迁移的问题中，现实环境作为目标域，其具体的物理特性是目前技术难以完全模拟的。为了解决这个问题，产生了一类基于物理环境[7,8]的随机化方法，这类方法首先使用随机的动力学相关参数，包括机器人每一部分的质量、关节接口处的阻尼、被操作物体的质量、物体和桌面的摩擦系数、桌子的高度、位置传感器的数据、两次动作间隔的时间步等。对于这些参数进行随机化的目的是缩小仿真环境在动力学参数方面和现实环境的差异，从而提高任务的成功率。而本文提出的方法则考虑到仿真环境无论如何不可能和现实环境完全一致，目标驱动的算法中存在类似人类操作过程中的目标(goal)。将这个目标(goal)在操作空间范围内进行符合特定随机化分布的处理。通过随机化目标(goal)，来使得训练出的策略对于任务执行过程中达到的结果和预计目标的差异有更好的兼容性。

基于环境视觉[9, 10]的随机化方法则使用了随机化的视觉信息，包括杂物的形状和数目、待操作物体的纹理和位置、桌子和地板等背景纹理、模拟相机的位置和方向、背景光源的数量和位置、物体表明对光反射的性质等。对于这类数据随机化则是考虑到智能体观察世界的方式有限，无法获得物体的材质等信息，而这些信息对于后面的训练效果又有着很大的影响。这类方法在解决仿真到现实迁移的问题上的本质思路与动力学随机化相似，也是通过对于数据集的扩展来缩小仿真环境和现实环境的差距，目前来看这种差距是无法完全消除的。

本文对目标驱动(goal-conditioned)强化学习中的目标进行了随机化，目标(goal)本身是这个过程中特有的状态，和以往随机化的环境参数有着本质的不同。对于目标的随机化是充分考虑了目标驱动的算法特点，其现实意义体现在：机器人操作本身就是一个欠驱动的系统，对于系统实际控制的结果往往和模型预期的有差距，这个差异不完全是由仿真系统的物理参数和现实不同造成的，驱动器的运动策略等内容也会造成差异，使用精度更高的驱动器可以缩小这个差距但是无法完全抹除，因此对于目标的随机化可以让学习到的策略适应这种欠驱动导致的误差，从而提高操作的成功率。此外，因为仿真环境对于物理参数模拟的误差通常也会和系统的欠驱动性所叠加导致最终目标的误差，因此对于目标的随机化方法同样对于适应物理参数的误差有较好的效果。

2.2 目标随机化方法

HER 算法天然就是一种目标驱动(Goal-conditioned)的算法，因此可以通过对训练中的目标(goal)进行处理来提高得到策略的泛用性。在训练的时候，每一段(episode)开始时，都对其中的目标 g 进行随机化处理，加入符合特定分布的随机化参数来模拟上述情况。具体的操作为：根据每个从轨迹中采样的状态转移(transition)生成一个新的目标(new goal)，鉴于本文研究的内容为机器人的操作任务，文中的目标为被操作物体的坐标，最终的目标就是操作任务需要将物体操作到的终点坐标(对于开门任务，门把手轴的中点坐标可以作为目标)。新的目标的选取采用了对于未来状态的预测(即将某个状态转移所在轨迹中若干步骤后的状态作为该状态转移的新目标)。

加入目标随机化参数策略的目的是：利用特定的方式对生成的新目标进行随机化处理(因为文中操作任务的目标均为操作物体的状态坐标，随机化处理的方式就是将目标中包含的坐标值乘以一个随机系数)，随机化处理后的目标在任务空间范围内相对于原始的目标应当符合特定的分布。

最后将随机化处理过的目标和状态转移组合并存放入缓冲区中，利用缓冲区列表中的数据进行训练。经验回放的部分使用 DDPG 这个针对连续行为的策略学习方法作为离线策略算法。下面给出了仿真环境下采样训练以及将目标进行域随机化操作的过程，具体流程如算法 2 所示。

算法 2 基于目标的域随机化算法

输入：离线强化学习算法 A；目标随机化参数策略 $\mathbb{X}$ ；奖励函数 r；
输出：训练好的策略网络
初始化 A，回放缓冲区 R
for episode in range
获得初始状态 s_0 和最终目标 g
for t in range (0, T-1)
根据当前状态 s_t 和 g 计算出下一步的动作 a_t
执行动作 a_t 得到新的状态 s_{t+1}

```
    end for
    for t in range (0, T-1)
            获得目标 g (HER)
        产生随机化系数 𝕏
        G←g×𝕏
        for g' in range G
            r'=r(s_t, a_t, g')
            将轨迹 (s_t || g', a_t, r', s_{t+1} || g') 存入 R
        end for
end for
    利用 A 和 R 优化策略
end for
```

算法 2 的主要操作在于迭代部分对于目标的随机化操作，在每次迭代中首先使用经验回放方法(HER)获取目标 g，然后根据操作任务的任务空间(空间中驱动器与待操作物体的分布范围)和随机化系数(如高斯分布函数)，将目标随机化为集合 G，这样得到的集合 G 相对初始的 g 在任务空间内满足随机化系数的概率分布，同时集合 G 又受到任务空间的约束，防止产生过大或过小的数据影响训练。通过上述对 g 的随机化操作，得到的集合 G 可以一定程度反映欠驱动操作系统的误差，提高算法迁移到现实环境之后的成功率。最后将集合中的随机化元素 g' 和采样得到的轨迹 (s_t, a_t) 结合，存入缓冲区用于优化策略。

3　实验与性能评估

本节通过设置在仿真环境下和现实中两个部分的对比实验，来评估本文所提出的算法仿真任务中的成功率，训练收敛速度，任务参数变化后算法的泛用性，以及迁移至现实环境后任务的成功率。

3.1　环境配置

本次实验中，仿真环境搭建在 Ubuntu 系统下，使用了 Pytorch 来进行网络的搭建，使用了 Mujoco[18]作为物理引擎，为了保证训练出的策略可以在现实环境中测试，本文在仿真环境中对现实中的环境进行了建模，包括机械臂(UR5e)[19]，电动夹爪，实验的平台和部分物体，仿真环境中的场景如图 1 所示。

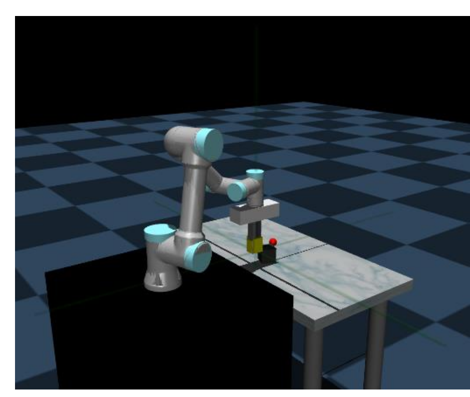


图 1　Mujoco 环境下机器人操作的可视化

Fig. 1　Visualization of robot manipulation in Mujoco

本文主要通过三个不同的任务场景来评估算法的性能。

1)推箱子(Pushing)：这个任务的场景包含一个桌子，放置于桌面的物块，和一个机械臂。任务的目标是使用机械臂将物块推至桌子上规定的位置。由于整体任务内容不涉及抓取，任务过程中末端的夹爪状态始终处于闭合。

2)抓取(Pick-and-place)：这个任务的场景和推箱子类似，不同之处在于任务的目标是使用夹爪将物块夹取至空中的指定位置。整体任务内容涉及到了抓取，所以包含了对于夹爪开合的控制。为了保证训练的成功率，实验中将夹爪夹住物块的状态设置为任务的初始状态。

3)开门(Door-opening)：这个任务相较于前两个任务复杂了许多，场景中包含一扇门框固定的门和一个机械臂。任务的目标是使用夹爪握住门把手，向下扳动把手将门打开，再将门拉开到一定角度。由于整体任务十分复杂，本文将夹爪握持住门把手的状态，设置为任务的初始状态。

采样中的状态，包括了 Mujoco 环境中的机械臂夹爪末端的位置信息，还有环境中所有物体的坐标。

对于目标在采样中如何进行随机化，本文的目标随机化参数选用的是符合正态分布的随机数。即，对离线策略训练的时候采样中的目标(goal)，对其进行了符合 N(0，0.1)分布的随机化变化。

3.2　仿真结果分析

为了评估本文提出的算法的性能，此处与一般强化学习(RL)算法[20]，事后经验回放(HER)算法[12]和动力学随机化(Dynamics Randomization)算法[7]三种强化学习算法进行比较。其中传统的强化学习算法。

1)成功率

为了衡量不同方法之间的性能差距，本文在上述的三个仿真环境内对几个算法进行训练。在测试算法成功率的时候，为了评估不同方法对环境的适应性和鲁棒性，本文将测试中不同任务的初始环境都加入了随机的扰动。下面统计了 50 次本文将训练和测试时环境变化的程度比值定义为初始状态随机系数，并在下面的测试中将初始状态随机系数设置为 1.5。表 1 展示的是不同算法在不同实验环境中的成功率差距。

表 1　不同操作任务中的成功率比较

Tab. 1　Comparison of success rate on dexterous manipulation tasks

Method	Pushing	Pick&Place	Door-Opening
RL	56%	10%	6%
HER	64%	38%	10%
Dynamic Random	84%	42%	34%
Goal-based Random	80%	64%	66%

从结果可以看出，即使经过了细节的改进，一般强化学习算法在面对较大的搜索空间时表现也比较一般。事实上，复杂任务如果不设置初始状态则一般强化学习的成功率基本为 0。经验回放方法解决了搜索空间的问题，但是当初始状态发生随机改变之后并不能很好的适应。物理量域随机化的方法与本文提出的基于目标的域随机化方法都对经验回放进行了改进，均有着更好的效果，并且本文的方法基于目标分析，对于复杂任务的适应性更强。

为了更进一步的比较几个算法之间在不同任务间泛用性能的差距，此处进一步提高初始状态的随机系数，表 2 展示的是初始状态随机系数提高至 2.5 后不同算法的成功率。

表 2　更高初始状态随机系数下的成功率比较

Tab. 2　Comparison of success rate on higher initial state randomness

Method	Pushing	Pick&Place	Door-Opening
RL	22%	0%	2%
HER	60%	44%	14%
Dynamic Random	58%	38%	16%
Goal-based Random	76%	52%	32%

2)收敛时间

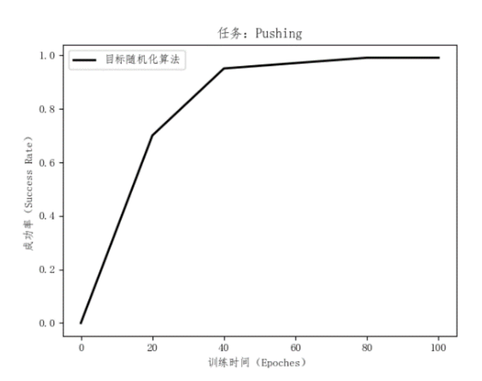
为了衡量算法的训练速度，本文还将评估算法在训练中收敛所需要的时间，图 2 展示的是本文所提出的算法在不同任务中收敛所需要的训练时间。

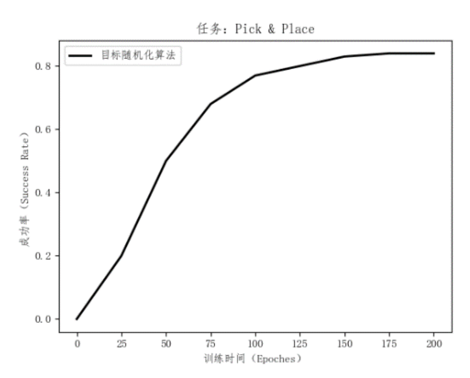
3.3　现实环境验证

为了在现实环境中验证前面训练得到的策略的可靠性，本次实验除了在硬件设备方面使用了和仿真环境中相同的配置(UR5e, Universal Robots)之外，为了准确获取场景中物体的

坐标，现实环境下的机械臂如图 3 所示

为了控制机械臂运动，实验中主机和机械臂使用 Python 环境下的实时控制，以局域网广播的方式进行通信和控制。夹爪的张开角度由夹爪本身的舵机驱动进行反馈，机械臂的末端位置则通过与机械臂本身的通信获取。





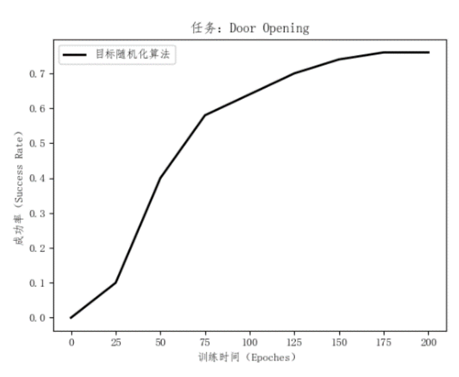


图 2　不同任务中的训练速度

Fig. 2　Training speed in different tasks



图 3　现实任务中使用的机械臂

Fig. 3　Manipulator used in reality

为了采集场景中其他物体的坐标，本次实验中使用了多摄像头系统(Multi-camera system，Opti-track)，通过在物体表面和边缘的关键位置粘贴标定点，可以采集到物体在环境中的坐标。对于开门任务来说，为了保证标定的准确性，在门框，门板，门把手上都粘贴了多组定位点，以克服门本身对于摄像头的遮挡导致定位失效的情况。为了将 MCS 系统和机械臂本身的坐标系统一，在实验开始之前需要将机械臂固定。此外，也需要在机械臂的关键位置和末端粘贴标定点，将机械臂在 MCS 系统中的坐标进行标定，以方便将机械臂本身返回的坐标系映射到 MCS 的坐标系中。

本文在现实环境中对算法进行了测试，测试的时候主要执行了开门(Door Opening)的任务，并且对于门的初始位置也加入了初始状态的随机系数。实际实验中夹爪的初始位置和仿真环境中一样均已经夹住门把手。为了提高实验成功率，本次实验选用了有柔性夹取机构的机械夹爪，同时在夹爪与机械臂连接处的法兰盘使用了 3D 打印的结构，发生严重意外的时候只需要更换 3D 打印结构即可，防止错误的操作导致关键设备损坏。表 3 展示的是不同的初始状态随机系数下，任务的成功率。

表 3　现实环境中开门任务成功率

Tab. 3　Success rate of reality door-opening task

Initial rate	1.0	1.25	1.5	1.75	2.0
Success rate	92%	76%	60%	52%	24%

可以看出，在上述任务中现实环境中执行仿真环境下训练得到的策略时，当初始状态变化不大的情况下，有着较高的成功率。当初始状态变化较大的时候，相较于仿真环境中，现实实验的成功率会显著下降，其原因在于：门把手和门都是运动范围受限的刚体结构，机器人操作的误差会导致机械臂力反馈的急剧增大，触发机械臂的保护机制从而导致任务失败，若没有力反馈的保护机制则极容易在操作过程中损坏门和夹爪。

4　结束语

本文提出了基于目标的域随机化算法，通过将经验回放方法和基于目标的域随机化方法相结合，不仅在收敛速度方法表现良好，还可以更好的适应任务环境的变化，在任务初始状态变化较大的情况下仍旧取得了较好的表现，并且在将方法迁移至现实环境后也可以有着不俗的成功率。

本文的方法充分发挥了基于目标的算法的特性，基于目标的方法不仅能在奖励稀疏的环境中提高训练的效率，在使用了域随机化方法之后，还能提高在不同环境任务中的适应性和鲁棒性。于此同时，在策略从仿真环境迁移到现实环境的过程中，这种对于环境差异的适应性有效的提高了策略迁移的成功率。

但是现实中的具体操作任务，往往有着更加复杂的任务步骤，任务的目标也可能不仅仅是一个具体的坐标点因而更加难以定义，这些情况下基于目标的算法往往难以取得很好的效果。如何在复杂困难的情况中定义任务目标并选择合适的随机化方法是目前面临的主要困难。因此，尝试将本文提出的方法扩展到更多种类的机器操作任务中将是接下来工作的主要方向。

参考文献：

[1] Jens, Kober, J, *et al.* Reinforcement learning in robotics: A survey [J]. The International Journal of Robotics Research, 2013.

[2] Kroemer O, Niekum S, Konidaris G D. A review of robot learning for manipulation: Challenges, representations, and algorithms [J]. Journal of machine learning research, 2021, 22 (30) .

[3] Andrychowicz M, Baker B, Chociej M, *et al.* Learning dexterous in-hand

manipulation [J]. The International Journal of Robotics Research, 2020, 39 (1): 3-20.

[4] 张智广. 基于深度强化学习的机械臂抓取方法研究 [D]. 哈尔滨工业大学, 2021. (Zhang Zhiguang. Research on manipulator grasp based on deep reinforcement learning [D]. Harbin Institute of Technology, 2021.)

[5] Gupta A, Devin C, Liu Y X, *et al.* Learning invariant feature spaces to transfer skills with reinforcement learning [J]. arXiv, 2017.

[6] Chen X, Hu J, Jin C, *et al.* Understanding Domain Randomization for Sim-to-real Transfer [J]. arXiv, 2021.

[7] Peng X B, Andrychowicz M, Zaremba W, *et al.* Sim-to-real transfer of robotic control with dynamics randomization [C]// 2018 IEEE international conference on robotics and automation (ICRA) . IEEE, 2018: 3803-3810.

[8] Chebotar Y, Handa A, Makoviychuk V, *et al.* Closing the sim-to-real loop: Adapting simulation randomization with real world experience [C]// 2019 International Conference on Robotics and Automation (ICRA) . IEEE, 2019: 8973-8979.

[9] Tobin J, Fong R, Ray A, *et al.* Domain randomization for transferring deep neural networks from simulation to the real world [C]// 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS) . IEEE, 2017: 23-30.

[10] Niu H, Hu J, Cui Z, *et al.* DR2L: Surfacing corner cases to robustify autonomous driving via domain randomization reinforcement learning [C]// The 5th International Conference on Computer Science and Application Engineering. 2021: 1-8.

[11] Rusu A A, Večerík M, Rothörl T, *et al.* Sim-to-real robot learning from pixels with progressive nets [C]// Conference on Robot Learning. PMLR, 2017: 262-270.

[12] Christiano P, Shah Z, Mordatch I, *et al.* Transfer from simulation to real world through learning deep inverse dynamics model [J]. arXiv preprint arXiv: 1610. 03518, 2016.

[13] Sutton R S, Barto A G. Reinforcement learning: An introduction [M]. MIT press, 2018.

[14] Andrychowicz M, Wolski F, Ray A, *et al.* Hindsight experience replay [J]. Advances in neural information processing systems, 2017, 30.

[15] Ding Y, Florensa C, Abbeel P, *et al.* Goal-conditioned imitation learning [J]. Advances in neural information processing systems, 2019, 32.

[16] Dhiman V, Banerjee S, Siskind J M, *et al.* Learning goal-conditioned value functions with one-step path rewards rather than goal-rewards [J]. 2018.

[17] Lillicrap T P, Hunt J J, Pritzel A, *et al.* Continuous control with deep reinforcement learning [J]. arXiv preprint arXiv: 1509. 02971, 2015.

[18] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control, ”in IROS, 2012, pp. 5026–5033.

[19] 吴璞，夏长林，景鸿翔. UR5 机器人运动学分析与轨迹规划研究 [J]. 煤矿机械, 2021, 42 (04): 55-58. DOI: 10. 13436/j. mkjx. 202104018. (Wu Pu, Xia Changlin, Jing Hongxiang Research on kinematics analysis and trajectory planning of UR5 robot [J] Coal Mine Machinery, 2021, 42 (04): 55-58. DOI: 10. 13436/j. mkjx. 202104018)

[20] Schulman J, Wolski F, Dhariwal P, *et al.* Proximal policy optimization algorithms [J]. arXiv preprint arXiv: 1707. 06347, 2017.